B. J. McCarragher
Department of Engineering, the Faculties,
The Australian National University,
Canberra, ACT 0200 Australia

H. Asada
Department of Mechanical Engineering,
Massachusetts Institute of Technology,
Cambridge, MA

The Discrete Event Control of Robotic Assembly Tasks

A new approach to process modeling, task synthesis, and motion control for robotic assembly is presented. Assembly is modeled as a discrete event dynamic system using Petri nets, incorporating both discrete and continuous aspects of the process. The discrete event modelling facilitates a new, task-level approach to the control of robotic assembly. To accomplish a desired trajectory a discrete event controller is developed. The controller issues velocity commands that direct the system toward the next desired contact state, while maintaining currently desired contacts and avoiding unwanted transitions. Experimental results are given for a dual peg-in-the-hole example. The experimental results not only demonstrate highly successful insertion along the optimal trajectory, but also demonstrate the ability to detect, recognize and recover from errors and unwanted situations.

1 Introduction

Task level process monitoring and control are keys to improving the success of robotic assembly. The focus of this paper is to present a new approach to the task-level control of robotic assembly, treating assembly as a discrete event dynamic system. The abstraction to discrete event modelling highlights the necessary transitions for successful assembly. Conditions on the control action can then be developed which cause the necessary transitions to occur. Moreover, the abstraction allows for planning on a task level rather than the cumbersome process of exact trajectory planning. Lastly, a discrete event approach provides the means to recognize and recover from unwanted situations. This contingency performance is one of the significant advantages to this approach.

A discrete event system is a dynamic system in which the state vector is discrete. Due to the discrete nature of the state vector, state changes occur at discrete points in time in response to the occurrence of certain events. Typically, discrete event dynamic system models arise from certain aspects of manufacturing systems and data network protocols. In contrast to these applications, this paper presents the discrete event modeling and control of robotic assembly tasks. A discrete event in an assembly task is defined as a change in the state of contact. At this instant, changes in the dynamic equations of motion describing the process are significant. As well, the specified control commands must be changed to accommodate the change in dynamics. The changes in both the dynamics and the control of assembly processes precisely at the point when the discrete state of contact changes serve as the motivation for the discrete event modelling of assembly.

Various methods have been proposed for the control of assembly processes. Whitney [18] derives quasi-static conditions for a successful insertion operations. Mason [7] and Raibert and Craig [13] describe a hybrid position/force control for various manipulation tasks. Other work has included using force feedback as a source of task performance information [5, 3]. However, unlike most force-feedback applications, the state of contact changes during an assembly process. These discrete changes in state should be the focal point of the control strategy since they indicate the significant changes in the system dynamics. In order to incorporate the state changes, we will model assembly as a discrete event dynamic system.

The difficulty of modelling and designing real-time discrete event control systems has long been recognized in the literature. Moreover, there is apparently no unifying theory for the control of discrete event systems. Numerous methods have been proposed for the modelling and analysis of discrete event systems, each having different characteristics. Ostroff and Wonham [11] provide a powerful and general framework (TTM/RTTL) for modelling and analyzing real-time discrete event systems. Holloway and Krogh [4] use cyclic controlled marked graphs (CMG's) to model discrete event systems, allowing for the synthesis of state feedback logic. Other approaches include formal languages [14, 15], finite state machines [16], and Petri nets [12]. Most of this work is concerned with developing and proving control-theoretic ideas for specific classes of systems.

The goal of this paper is to present and demonstrate the successful discrete event control of robotic assembly tasks. The control system, represented by the block diagram in Fig. 1, is a hierarchical configuration consisting of two primary components. The first component is required to recognize and monitor the process state. In this paper we will use the process monitor developed in [9]. This effective and fast process monitor is based upon qualitative dynamic process models to detect changes in discrete states of assembly. The second component is the discrete event controller, which is the focus of this paper.

In this paper, a discrete event model of the assembly process is developed. Currently, only planar processes are considered. Once the discrete event model is proposed, a discrete event controller, or DEC, is developed. The DEC is a task-level controller that directs the assembly process through a series of events to the desired end state. First, a discrete event trajectory, or path through the discrete states, is specified [10]. The trajectory specifies the discrete controls which control the discrete transitions. The discrete control vector is then mapped into continuous velocity commands to be sent to the robot arm. The execution of these velocity commands causes the system to pass through the desired events. The applicability of the approach is demonstrated experimentally. Results are given for both an optimal trajectory and for contingency trajectories. That is, the system demonstrated the ability to detect, identify and recover from unwanted situations.

2 Discrete Event Modeling

This section summarizes a new way of describing the discrete event nature of robotic assembly using Petri nets which is fully described in [10]. Petri nets are a compact mathematical way of describing the geometric constraints and the admissible transitions for an assembly task. Moreover, Petri nets are a useful method for describing the indeterministic nature of robotic assembly, by incorporating transitions that are possible given the

Contributed by the Dynamic Systems and Control Division for publication in the JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL. Manuscript received by the DSCD August 11, 1992. Associate Technical Editor: R. Shoureshi.

uncertainties, unknowns and errors in the system. The ability to address these unknowns is one of the primary strengths of the Petri net modelling method.

A standard Petri net is composed of four parts [12]: a set of places $\mathcal{P}$, a set of transitions $\mathcal{T}$, an input function $\mathcal{I}$, and an output function $\mathcal{O}$. In addition we define a set of discrete controls $\mathcal{B}$. The input function $\mathcal{I}$ is a mapping from the places to the transitions, while the output function $\mathcal{O}$ is a mapping from the transitions to the places. The number of discrete controls is identical to the number of transitions, and is denoted c. The number of places is p. Places can be thought of as conditions of the current state that enable the transitions to occur, or fire. Likewise, controls can be thought of as conditions of external inputs that enable the transitions to occur. The difference is that places are a function of the state of the system, whereas the controls are external inputs to the system chosen by the designer. It is important to note that the controls only allow or prevent a transition from occurring, they do not force the transition to occur.

The edges, or vertices, of both the workpiece and the constraint are denoted generally as ϕ. The facets, or surfaces, of both the workpiece and the constraint are denoted generally as ψ. In modelling the assembly process with a Petri net, then, we define a place to represent one contact pair (ϕ, ψ); either a surface of the workpiece in contact with an edge of the fixture or an edge of the workpiece with a surface of the fixture. Essentially, this means that each place represents only one constraint equation. Combinations of places can be used to describe cases of two-point contact where two constraints are simultaneously active. To make the description complete there is also a place modelling the condition of no-contact, that is, the null constraint equation.

Consistent with the definition of a place, we define a transition as the gaining or losing of a single contact pair, or constraint. Therefore, the occurrence of a transition is a discrete event, or change in contact state. The input function defines the places that must be active for a given transition to fire; that is, the contact pairs necessary for a given change of contact to occur. When the place conditions for a given transition are met, the transition is said to be place enabled. The output function defines the contact pairs resulting from a discrete event.

The set of discrete controls are defined such that there is one control variable for each transition. The controls are binary valued, with a "one" indicating that the associated transition is enabled, and a "zero" indicating that the transition is disabled. A transition with a control variable of "one" is said to be control enabled. A transition that is both place enabled, as defined by the input function, and controlled enabled is simply referred to as enabled.

A marking γ of a Petri net is a $(p \times 1)$ vector assignment of tokens to the places of the net, where p indicates the number of places; that is, the total number of possible contact pairs. Tokens can be thought of residing in the places of the net. A token residing in a place indicates that the given edge and surface are in contact. That is, the constraint represented by that place is currently active. For our purposes here, a place can only have either zero or one token.

The execution of the Petri net is controlled by the discrete controls and the distribution of the tokens. If a token exists in each of the input places to a transition, that transition is said to be place enabled. A transition that is both place and control enabled may fire. A transition fires by removing the token from each of the input places and establishing a token in each of the output places. Note that a place may be both an input and an output of a single transition. The firing of a transition results in a next desired marking γ_d, describing the new state of contact represented by the distribution of tokens. By enabling various transitions and executing the net, we can direct the system through a series of discrete contact changes (events) to the desired final state.

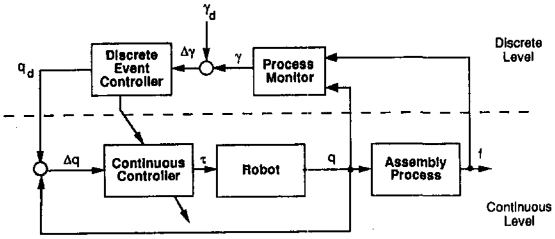

Fig. 1 Block diagram of discrete event control structure

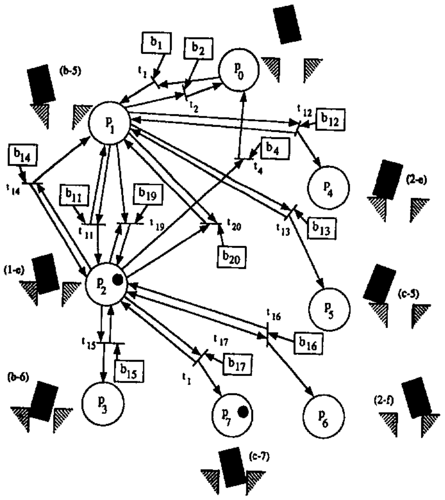

Fig. 2 Petri net model of single peg-in-the-hole

A Petri net can be represented graphically, with a circle ○ representing a place, a bar | representing a transition, a box □ representing a discrete control, and a bullet • representing a token. Directed arcs connect the places and the transitions. An arc from a place into a transition represents an input of that transition. An arc from a transition to a place represents an output of that transition. An arc from a control variable into a transition indicates the control for that transition.

A Petri net model for a limited portion of the assembly process is shown in Fig. 2. For this model, the set of places is given by

$$\mathcal{P} = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7\} \qquad (1)$$

where the places and their associated contact pairs are

$$\begin{array}{llll} p_0\text{: null} & p_2\text{: } (1-e) & p_4\text{: } (2-e) & p_6\text{: } (2-f) \\ p_1\text{: } (b-5) & p_3\text{: } (b-6) & p_5\text{: } (c-5) & p_7\text{: } (c-7) \end{array} \qquad (2)$$

Thus, the number of places is $p = 8$. The set of transitions is given by

$$\mathcal{T} = \{t_1, t_2, \ldots, t_{26}\} \qquad (3)$$

The number of transitions is $c = 26$. The set of controls is given by

$$\mathcal{B} = \{b_1, b_2, \ldots, b_{26}\} \qquad (4)$$

The marking shown in Fig. 2 is given by a $(p \times 1)$ vector:

$$\gamma = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]^T \qquad (5)$$

indicating that the system is in the two point contact case.

3 Assembly Constraints and Discrete Event Conditions

As was pointed out earlier, during assembly the state of contact changes. Each contact represents a constraint on the rigid

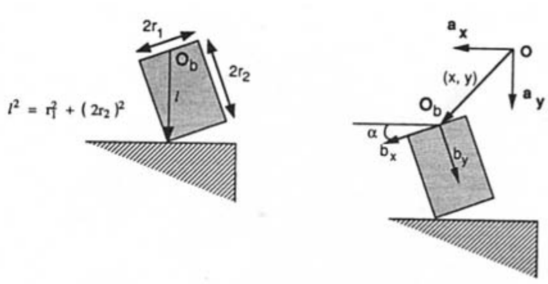

Fig. 3 Definition of generalized coordinates

body dynamics describing the motion of the manipulator and the workpiece. The changes in contact, therefore, result in different constraint equations. Changing constraint equations result in a varying number of motion degrees of freedom and a varying number of equations of motion. Using Petri net modelling, we can fully describe all the possible constraint equations resulting from the various states of contact with only one matrix equation. Additionally, for nominal part locations, we will derive a necessary condition for a discrete event to occur, called the enabling condition, and we will derive a sufficient condition for a discrete event not to occur, called the disabling condition.

We define an inertial coordinate system called Frame **a**, and we define Frame **b** to be attached to the workpiece as shown in Fig. 3. The generalized coordinates for planar rigid body motion are the x and y position in inertial space and the orientation α of Frame **b** in Frame **a**. The generalized coordinates will be expressed as a vector $\mathbf{q} = [x, y, \alpha]^T$.

We denote the position vector from the inertial origin to an arbitrary vertex ϕ as $\mathbf{d}_\phi$, as shown in Fig. 4. A surface ψ, on the other hand, is represented by the unit normal vector $\mathbf{n}_\psi$ and the position vector $\mathbf{d}_\psi$ as shown in the figure. For the sake of convenience, we define the normal vector to be an outward vector, and the position vector is one of the two vertices that bound the surface.

Let $h_{\phi\psi}$ be the distance between edge ϕ and surface ψ given by

$$h_{\phi\psi} = (\mathbf{d}_\phi - \mathbf{d}_\psi)^T \mathbf{n}_\psi \tag{6}$$

The contact between the edge and the surface is described simply by

$$h_{\phi\psi} = 0 \tag{7}$$

We wish to specify admissible robot velocity commands that will satisfy the geometric constraint (7). Note that one of the position vectors in (6) defines the motion of the workpiece, and can be written as a function of the robot position; $\mathbf{d} = \mathbf{d}(\mathbf{q})$. If the surface involved is part of the workpiece, then the unit normal vector is also a function of $\mathbf{q}$; $\mathbf{n}_\psi = \mathbf{n}_\psi(\mathbf{q})$. Thus, to derive admissible velocities that satisfy the geometric constraints, we simply differentiate equation (7).

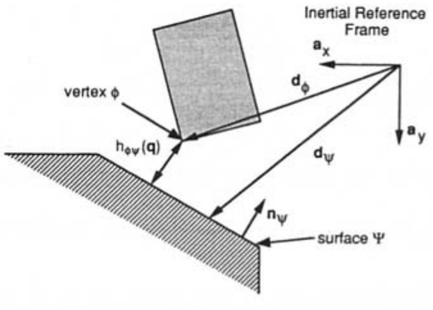

Fig. 4 Definition of geometric position

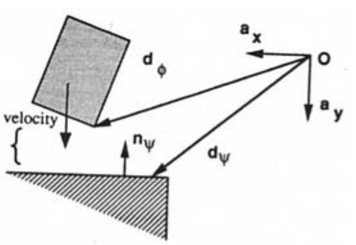

Fig. 5 Discrete event resulting in gain of contact

$$\frac{d}{d\mathbf{q}}\,[(\mathbf{d}_\phi - \mathbf{d}_\psi)^T \mathbf{n}_\psi]\,\frac{d\mathbf{q}}{dt} = 0 \tag{8}$$

Equation (8) describes the velocity vector that allows the system to move without violating the constraint. We can rewrite equation (8) to yield

$$\mathbf{a}_i \dot{\mathbf{q}} = 0 \tag{9}$$

where $\mathbf{a}_i$ is a 1×3 row vector given by

$$\mathbf{a}_i = \frac{d}{d\mathbf{q}}\,[(\mathbf{d}_\phi - \mathbf{d}_\psi)^T \mathbf{n}_\psi] \tag{10}$$

Note that $\mathbf{a}_i$ is the velocity coefficient vector describing the constraint represented by the place p_i.

The notion of marking γ used in the Petri net modelling will allow us to combine all the different constraint equations into one matrix equation. First, we define the constraint matrix **A** to incorporate the velocity constraint equations for all the places in the Petri net. Thus, **A** is a ($p \times 3$) matrix, where p is the number of places in the net. Using Eq. (9), **A** is given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_p \end{bmatrix} \tag{11}$$

Additionally, the Petri net marking indicates which of the constraints are active at a given instance. Therefore, we can describe the constraint equations for the entire assembly process with only one matrix equation.

$$\Gamma \mathbf{A} \dot{\mathbf{q}} = 0 \tag{12}$$

where $\Gamma = \operatorname{diag}\{\gamma_1, \ldots, \gamma_p\}$. Equation (12) gives a compact and flexible description of the overall geometric constraints encountered during the assembly process modelled by a Petri net. The Petri net marking γ is a type of switch that indicates which constraints are active and which are inactive at any given point in time.

In addition to determining motion that maintains a contact, it is desired to determine the motion such that the workpiece makes the next discrete change. Since we only allow one contact to be either lost or gained at any one instant, the next desired constraint to become active or inactive can be determined uniquely given the next desired marking γ_d as input. We again denote this general constraint with a contact pair as (ϕ, ψ), where ϕ is the edge of contact and ψ is the surface of contact. We now describe two cases, one for a gain of contact and one for a loss of contact.

Gain of Contact. Since the desired contact is not yet active, the current distance between edge ϕ and surface ψ is positive as shown in Fig. 5. To gain contact, the distance must decrease:

$$\frac{d}{dt}\,[h_{\phi\psi}(\mathbf{q})] = \mathbf{a}_i \dot{\mathbf{q}} < 0 \tag{13}$$

The change of marking at place p_i, $\Delta\gamma_i$, is given by

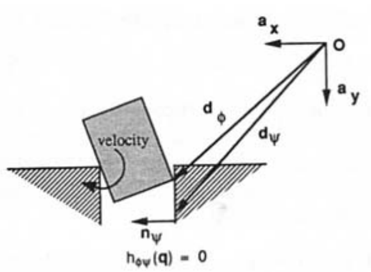

Fig. 6 Discrete event resulting in loss of contact

$$\Delta\gamma_i = 1 \tag{14}$$

Note that $\Delta\gamma_i$ is a scalar.

Loss of Contact. The second case is when the desired transition results in a loss of contact as shown in Fig. 6. In this instance, the distance between the edge and surface of the contact pair must increase. Therefore,

$$\frac{d}{dt}[h_{\phi\psi}(\mathbf{q})] = \mathbf{a}_i\dot{\mathbf{q}} > 0 \tag{15}$$

The change in marking γ_i is then

$$\Delta\gamma_i = -1 \tag{16}$$

Combining equations (13) through (16) yields

$$\Delta\gamma_i\mathbf{a}_i\dot{\mathbf{q}} < 0 \tag{17}$$

Equation (17) stipulates that any commanded velocity should cause the distance between the current position and the desired position to decrease. It is our enabling condition. It is a necessary equation for the transition specified by $\Delta\gamma_i$ to fire.

In addition to determining a condition for a discrete event to occur, it is desired to determine a condition for a discrete event not to occur. Since (17) is a necessary condition for a transition to occur, we can change the inequality sign to determine a sufficient condition for a transition not to occur. Hence, our disabling condition for a transition specified by $\Delta\gamma_j$ is

$$\Delta\gamma_j\mathbf{a}_j\dot{\mathbf{q}} \geq 0 \tag{18}$$

where $\Delta\gamma_j$ is the change in the marking for place p_j. That is, $\Delta\gamma_j = -1$ if the transition results in a loss of the contact pair represented by p_j, and $\Delta\gamma_j = 1$ if the transition results in a gain of the contact pair represented by p_j.

We will assume that an event trajectory has been specified according to the method described in [10]. The discrete controls **B** are easily determined, then, given the desired event trajectory. Those transitions that are enabled (17) on the discrete event trajectory have discrete control values of one, while all other place enabled transitions which are desired not to occur (18) have control values of zero. Transitions that are not place enabled are immaterial.

4 Discrete Control to Continuous Velocity Map

Once the discrete controls have been determined, it is desired to calculate the continuous control commands that will direct the system to follow the desired discrete event trajectory. The continuous control commands are the velocity commands actually sent to the motors of the robot. Essentially, we want to map the discrete control variables **B** into the continuous control variables $\dot{\mathbf{q}}$. Note, however, that a given **B** may not have a feasible solution. If a map does exist between **B** and $\dot{\mathbf{q}}$, it will usually be multi-valued.

4.1 Optimal Velocity Solution. The discrete event trajectory specifies that only one change of contact should occur at each stage of the process. This is a reasonable strategy for assembly since it is simpler and more robust than attempting two contact changes simultaneously. The necessary condition on the velocity command for the desired transition t_i to occur was given earlier by the inequality (17).

A significant feature of the optimal velocity solution is that the discrete event trajectory specifies that all changes of contact not on the trajectory are desired not to occur. That is, we wish to disable the other transitions. For a single transition, the disabling condition is given by the inequality (18), which is a sufficient condition to disable a transition, that is, for a transition not to occur. Thus, the conditions to disable a set of transitions resulting in $\Delta\gamma_j \neq \Delta\gamma_i$ is given by

$$\Delta\gamma_j\mathbf{a}_j\dot{\mathbf{q}} \geq 0 \quad \forall j \neq i \tag{19}$$

Equations (17) and (19) describe a set of inequality constraints on the continuous control vector $\dot{\mathbf{q}}$. If these constraints are met, the specified velocity command will effect the desired discrete event, that is, the desired change of contact state will occur.

The problem has now been reduced to solving a set of simultaneous linear inequalities (17) and (19). The solution to these simultaneous inequalities may not be unique. Therefore, we would like to determine a single, optimal solution. An optimal continuous control command that satisfies the inequality constraints can be determined by solving a linear optimization problem with linear constraints in the command velocities. The performance function to be optimized is selected to increase the reliability and robustness of the solution. In this case, we will adopt the Max-Min strategy, that is, maximize the minimum distance to each of the constraint equations given by inequalities (17) and (19).

$$J = \max_{\dot{\mathbf{q}}}\,[\min_{i\in\mathcal{A}} \mathbf{a}_i\dot{\mathbf{q}}] \tag{20}$$

where $\mathcal{A}$ is the set of indices included in all the inequalities of (17) and (19), and $(\mathbf{a}_i\dot{\mathbf{q}})$ is the distance between the velocity command given by $\dot{\mathbf{q}}$ and the constraint $\mathbf{a}_i$. Equation (20) is maximum at a point that is farthest from all the constraints, thus giving as large a margin for error as possible. Note that the constraint inequalities all go through the origin. Thus, the extreme solution will be unbounded. Since we are only concerned with the direction of the velocity commands and not the magnitude, we will require an additional condition to provide an upper bound on the cost function. The condition

$$\dot{x}^2 + \dot{y}^2 + (l\dot{\alpha})^2 = 1 \tag{21}$$

is imposed to limit the magnitude of the velocity vector, where l is the distance from the point of contact to the origin of Frame **b**, the point where the rotational velocity command is applied as shown in Fig. 3. Thus, we have translated the rotational velocity $\dot{\alpha}$ into a tangential velocity $l\dot{\alpha}$ for a consistent optimization process. Figure 7 gives a graphic depiction of the optimiza-

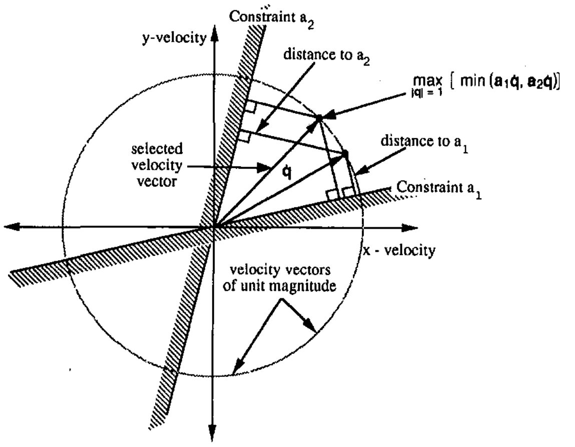

Fig. 7 Graphic description of the nonlinear optimization problem

Table 1 System parameters for example calculations

Parameter	Value
x	2.0 cm
y	2.0 cm
α	10 deg
r_1	1.0 cm
r_2	1.5 cm

tion routine (2-dimensional; no rotation). Hence, the maximum of Eq. (20) subject to the inequality conditions of Eqs. (17) and (19) gives us the continuous velocity control commands that satisfy the discrete event control commands. Note that since $\mathbf{a}_i$ is dependent upon the position and orientation of the workpiece, an optimization is required at various values of $\mathbf{q}$. Satisfying the discrete event control commands means that we have successfully determined the commands necessary to effect the desired trajectory in event space to complete the assembly task.

The problem as posed may not have a feasible solution. Best and Ritter [2] give a simple means to determine if a feasible solution exists. If a feasible solution does not exist, at least one of the constraint equations must be relaxed or removed. Removing a constraint equation no longer guarantees that the desired event will occur since other transitions are enabled. A situation with an infeasible solution can arise because the Petri net is an indeterministic model of the assembly task. Ranges of uncertainties and tolerancing errors may be too large to guarantee that a desired change of contact will occur. Additionally, Eq. (17) is only a necessary and not a sufficient condition for a transition to occur. The following example gives one case where the initially posed problem is infeasible.

4.2 Example of Discrete to Continuous Control Map. In this section we will determine the discrete controls and continuous velocity commands necessary for the path shown in Fig. 8. For the example given here we will only consider the transitions that are place enabled. The system parameters used for the calculations are given in Table 1.

The desired event trajectory requires the firing of three transitions, (t_{11}, t_{19}, t_{15}), as specified by the Petri net model. That is, three discrete changes of contact have to occur. Let us consider these events one at a time.

Event 1: Transition t_{11}. To fire transition t_{11}, we need to have the transition control enabled, $b_{11} = 1$. Additionally, we would like to maintain the current contact, place 1. Transition t_2 is a loss of the contact represented by place 1; as such $b_2 = 0$. Lastly, we would like to insure that no other place enabled transitions occur. According to the Petri net model, the two place enabled transitions are t_{12} to place 4, and transition t_{13} to place 5. Thus, $b_{12} = 0$ and $b_{13} = 0$. Using equation (17) for b_{11} and using equation (19) for b_2, b_{12} and b_{13}, the following four inequalities define the velocity commands for event 1.

$$\mathbf{a}_2\dot{\mathbf{q}}: [\cos(\alpha) \quad \sin(\alpha) \quad -x\sin(\alpha) + y\cos(\alpha)]\dot{\mathbf{q}} < 0 \quad (22)$$

$$\mathbf{a}_1\dot{\mathbf{q}}: [0 \quad 1 \quad r_1\cos(\alpha) - 2r_2\sin(\alpha)]\dot{\mathbf{q}} \geq 0 \quad (23)$$

$$\mathbf{a}_4\dot{\mathbf{q}}: [-\sin(\alpha) \quad \cos(\alpha) \quad -x\cos(\alpha) - y\sin(\alpha)]\dot{\mathbf{q}} \geq 0 \quad (24)$$

$$\mathbf{a}_5\dot{\mathbf{q}}: [0 \quad 1 \quad -r_1\cos(\alpha) - 2r_2\sin(\alpha)]\dot{\mathbf{q}} \geq 0 \quad (25)$$

Using a simplex method of optimization and the values given in Table 1 the following equation was solved.

$$J = \max_{\dot{q}}\,[\min(-\mathbf{a}_2\dot{\mathbf{q}}, \mathbf{a}_1\dot{\mathbf{q}}, \mathbf{a}_4\dot{\mathbf{q}}, \mathbf{a}_5\dot{\mathbf{q}})] \quad (26)$$

The optimal velocity vector with which event 1 will occur is

$$[\dot{x}, \dot{y}, l\dot{\alpha}]^T = [-0.9988, -0.0210, -0.0453]^T \quad (27)$$

This command causes the workpiece to move toward the hole in a negative x-direction, with a slight rotation in the counterclockwise direction. Because of the slight negative rotation, an upward motion in the negative y-direction is required to maintain contact with the surface.

Event 2: Transition t_{19}. For event 2 we wish to maintain the contact represented by place 2, while losing the contact represented by place 1. Clearly, to effect transition t_{19} we require $b_{19} = 1$. According to the Petri net model, the only other place enabled transition is t_{20}, which would result in a loss of the contact represented by place 2; thus $b_{20} = 0$. Applying (17) for b_{19} and (19) for b_{20} we derive two inequalities for event 2.

$$\mathbf{a}_1\dot{\mathbf{q}}: [0 \quad -1 \quad -r_1\cos(\alpha) + 2r_2\sin(\alpha)]\dot{\mathbf{q}} < 0 \quad (28)$$

$$\mathbf{a}_2\dot{\mathbf{q}}: [\cos(\alpha) \quad \sin(\alpha) \quad -x\sin(\alpha) + y\cos(\alpha)]\dot{\mathbf{q}} \geq 0 \quad (29)$$

(30)

The maximum solution to equation (20) with this set of inequalities is, again using the simplex method and the parameters given in Table 1,

$$[\dot{x}, \dot{y}, l\dot{\alpha}]^T = [0.0456, 0.99, -0.1335]^T \quad (31)$$

This velocity vector has the x, y and α velocities in the proper ratio to maintain the contact (1-e), while losing the contact (b-5). Essentially, this results in the workpiece going downward in the positive y-direction as fast as possible, which then results in a positive x-velocity and a negative α-velocity to maintain the desired contact.

Event 3: Transition t_{15}. We wish to complete the insertion through the occurrence of event 3. Event 3 requires transition t_{15} to fire, therefore $b_{15} = 1$. The Petri net model shows us that four other transitions are place enabled. In order to disable these by the discrete controls, we require $b_4 = b_{14} = b_{16} = b_{17} = 0$. The inequalities for this situation are

$$\mathbf{a}_3\dot{\mathbf{q}}: [1 \quad 0 \quad -r_1\sin(\alpha) - 2r_2\cos(\alpha)]\dot{\mathbf{q}} < 0 \quad (32)$$

$$\mathbf{a}_2\dot{\mathbf{q}}: [\cos(\alpha) \quad \sin(\alpha) \quad -x\sin(\alpha) + y\cos(\alpha)]\dot{\mathbf{q}} \geq 0 \quad (33)$$

$$\mathbf{a}_1\dot{\mathbf{q}}: [0 \quad 1 \quad -r_1\cos(\alpha) + 2r_2\sin(\alpha)]\dot{\mathbf{q}} \geq 0 \quad (34)$$

$$\mathbf{a}_6\dot{\mathbf{q}}: [-\sin(\alpha) \quad \cos(\alpha) \quad -x\cos(\alpha) - y\sin(\alpha)]\dot{\mathbf{q}} \geq 0 \quad (35)$$

$$\mathbf{a}_7\dot{\mathbf{q}}: [1 \quad 0 \quad r_1\sin(\alpha) - 2r_2\cos(\alpha)]\dot{\mathbf{q}} \geq 0 \quad (36)$$

Due to a conflict between inequalities (34) and (35), there is no feasible solution to the set of inequalities as posed. The reason for this conflict lies in the information that is known about the geometry of the system. Since we have assumed limited knowledge, we were forced to use inequalities as necessary conditions for a transition to occur or as sufficient conditions for a transition not to occur. Thus, we have taken a conservative approach to the transition represented by (35). Unfortunately,

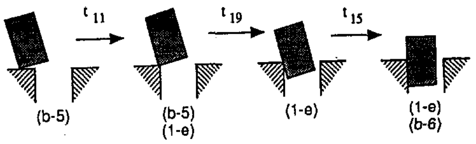

Fig. 8 Event trajectory of example velocity calculation

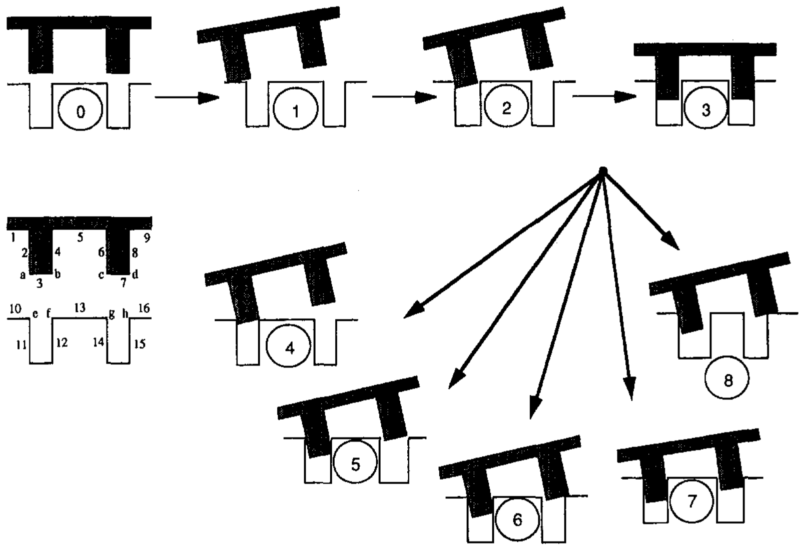

Fig. 9 Geometric model of dual peg-in-the-hole insertion for experiments

stating the problem in this manner does not allow for a feasible solution. One of the two inequality conditions must be removed. Since (34) is required for the workpiece to slide into the hole instead of out of the hole, (35) must be removed.

With (35) removed, the solution to (20) is given by the following velocity vector.

$$[\dot{x}, \dot{y}, l\dot{\alpha}]^T = [-0.2546, 0.9329, 0.2549]^T \qquad (37)$$

This solution effectively rotates the peg into position while maintaining the desired contact. Note that this solution does not guarantee that the system will not transit t_{17} resulting in a marking of places 2 and 7, rather than the desired transition t_{15} resulting in a marking of places 2 and 3 (see Fig. 2).

The actual continuous trajectory to be followed by the workpiece can be determined by integrating the optimal velocity for each configuration $\mathbf{q}$. This is to be done for events 1 through 3.

5 Implementation and Experiments

In an effort to demonstrate the feasibility of the discrete event control portion of the assembly system, a dual peg-in-the-hole insertion was conducted. A dual peg problem was selected to effectively demonstrate the method since a dual peg problem has significantly more potential contact states than a single peg problem. The task-level model of the system is shown in Figure 9. Initially, a simple event trajectory was chosen through states 0-1-2-3.

A four degree-of-freedom cartesian robot was used. However, the motion was kept planar with only two translations and one rotation being used. Correspondingly, only two forces and one moment were sampled for discrete event recognition [9]. The force data was taken using a six-axis foil strain gage sensor. A Dell 325 computer was used for robot control and data acquisition and analysis. The two forces and one moment signals were sampled at 300 Hz. The top speed of the robot was 80 cm/sec. The workpiece, which was an aluminum casing with unchamfered legs, was mated with a rigid aluminum environment with unchamfered holes. The clearance was 0.8 mm. The initial position and orientation of the workpiece relative to the constraint geometry is shown in Fig. 10.

The workpiece data use for the calculations of velocities is given in Table 2. The transitions and contacts that are significant are given in Table 3. Note that although only a limited portion of transitions are given, the full set of transitions were considered when the experiments were conducted.

5.1 Optimal Velocity Solution.

For the first transition 0-1, the best solution to avoid unwanted transitions is solely a y-velocity.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} 0.1^* \\ 1.0 \\ 0.0 \end{bmatrix} \qquad (38)$$

For the second transition, however, we wish to avoid the transitions that result in any of the following contact pairs

t_3	(a-10)(e-3)
t_4	(a-10)(b-10)
t_2	no contact

Thus, the optimal velocity commands are

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} 0.9935 \\ -0.0737 \\ 0.0865 \end{bmatrix} \qquad (39)$$

The slight rotation in the counter-clockwise direction better insures that transitions t_3 and t_4 do not occur; while the relationship among the three velocities maintains contact insuring that contact pair (a-10) is not lost. Lastly, the predominant velocity in the x-direction causes the desired transition t_5 (gain of contact (e-2)) to occur so that the system ends up in contact state 2.

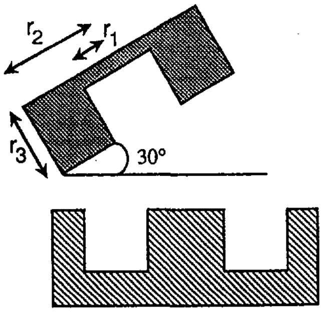

Fig. 10 Initial configuration of workpiece

Table 2 Workpiece data for dual peg-in-the-hole experiment

r_1	1.14 cm
r_2	2.40 cm
r_3	9.72 cm
α	10–40°

Table 3 Discrete event transitions for dual peg-in-the-hole experiment

Transition	From contact state	To contact state
t_1	no contact	(a-10)
t_2	(a-10)	no contact
t_3	(a-10)	(a-10) (e-3)
t_4	(a-10)	(a-10) (b-10)
t_5	(a-10)	(a-10) (e-2)
t_6	(a-10) (e-2)	(e-2)
t_7	(e-2)	(e-2) (a-10)
t_8	(e-2)	(e-2) (f-3)
t_9	(e-2) (f-3)	(e-2) (f-3) (b-12)
t_{10}	(e-2) (f-3) (b-12)	(e-2) (b-12)
t_{11}	(e-2)	(e-2) (h-7)
t_{12}	(e-2)	(e-2) (b-12)
t_{13}	(e-2)	(e-2) (d-15)
t_{14}	(e-2) (b-12)	(e-2) (b-12) (h-7)
t_{15}	(e-2) (b-12)	(e-2) (b-12) (d-15)
t_{16}	(e-2) (b-12) (h-7)	(e-2) (b-12) (h-7) (d-15)
t_{17}	(e-2) (b-12) (h-7) (d-15)	(e-2) (b-12) (d-15)

For the third transition 2-3, there are several unwanted contact pairs.

t_8	(f-3)
t_{11}	(h-7)
t_{12}	(b-12)
t_{13}	(d-15)
t_7	(a-10)

However, an initial attempt at solving the optimization will show that there is no feasible solution with the constraints as posed. This situation arises from the conservative approach to the robustness of the solution. The conditions used to derive the mathematical constraints are based on a sufficient condition for a transition not to occur, the so-called disabling condition (18). Examination of the constraint values at the best solution shows a conflict with transition t_{11}. Thus, for a feasible solution to be found, transition t_{11} should be removed since it is the only one not satisfied. With transition t_{11} removed, the optimal velocity vector is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} 0.4113 \\ 0.7153 \\ -0.5637 \end{bmatrix} \tag{40}$$

The significant feature of this velocity vector is the relatively large rotation component which causes the system to avoid most transitions bar the one that was removed from the optimization. Note also that the relationship among the velocities maintains the desired state of contact.

Samples of the resulting force profile and contact state recognition are given in Fig. 11 and Fig. 12. Figure 11 shows the results at a speed of 37 cm/s, while Fig. 12 shows the results at a speed of 64 cm/s. The optimal velocity vector proved to be successful for all transitions within the optimal discrete event trajectory. It also proved to be successful at both fast and slow speeds.

5.2 Contingency Performance and Error Recovery. To test the error recovery and contingency capabilities of the Petri net approach, a dual peg-in-the-hole experiment was again conducted. The desired optimal trajectory was as given in Fig. 9. The third transition of the optimal trajectory could not be guaranteed because the constraint associated with transition t_{11} was not satisfied. The velocity commands were determined assuming a relative angle of 30 deg, the same as in Section 5.1. However, the actual orientation has an allowed error. To test the error recovery capability, the system was run at several angles to cause an unwanted situation to occur. The optimal velocity commands were calculated and stored for the possible contact states not on the optimal event trajectory in accordance with the method described in Section 5.1.

The contingency experiments proved to be highly successful at all speeds. The velocity commands were determined using a relative angle of 30 deg. Relative angles of between 20 and 45 deg were tested resulting in different event trajectories. In each of the cases, the system was able to recognize the undesired state transition and to send the velocity commands that allowed

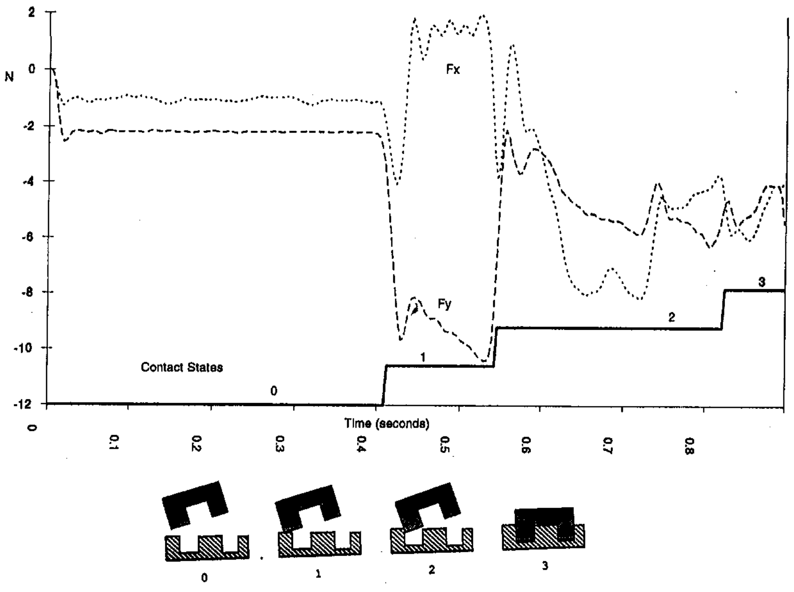

Fig. 11 Force profile and contact state recognition for dual peg-in-the-hole insertion using optimal velocity vector—slow speed (37 cm/s)

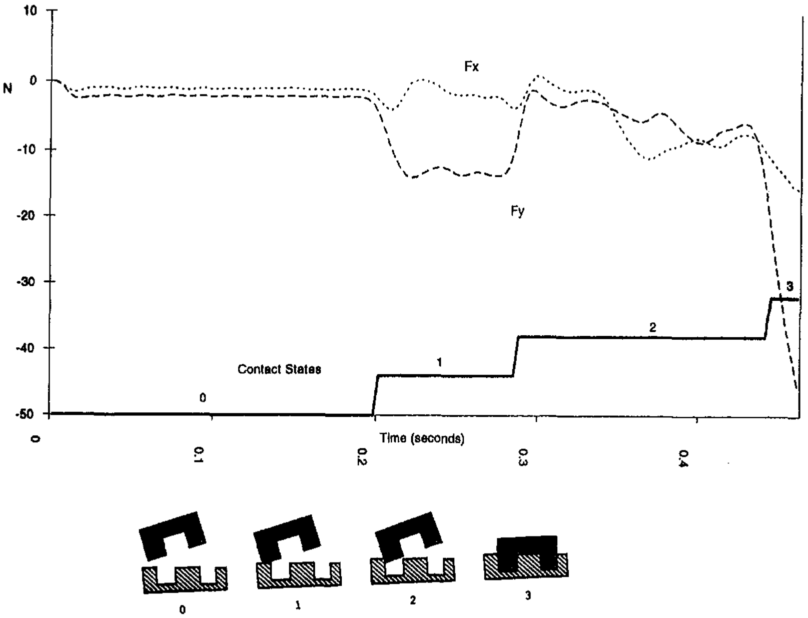

Fig. 12 Force profile and contact state recognition for dual peg-in-the-hole insertion using optimal velocity vector—fast speed (64 cm/s)

the system to recover from the misalignment. Figure 13 shows the force profile and contact states for a relative angle of 30 deg, which was the angle used to determine the optimal velocity commands. Note that the insertion was successful along the optimal trajectory. By contrast, Fig. 14 shows the force profile and contact states for a relative angle of 45 deg. In this case, the optimal velocity commands were not sufficient to avoid contact state 4. Once contact state 4 was reached, new optimal velocity commands were issued for the event trajectory 4-5-3. Again, however, due to the previous errors, the last transition was not successful. Instead, contact state 6 was reached. The system was again reconfigured such that the new event trajectory was 6-7-3. New velocity commands were calculated and issued which finally resulted in the successful insertion.

Another set of contingency experiments was run for the case where a misalignment occurs due to error in the robot position. Although the planned trajectory included state 1, the misalignment caused the system to go directly from state 0 to state 2. Again, the system proved to be highly successful at several speeds. Figure 15 shows the force profile with the contact state for this contingency situation run at a speed of 48 cm/s. Note that state 1 is not identified. Also, due to the initial misalignment, the optimal velocities were not successful. However, the contingency capabilities of the network were successful and proved to be robust.

The discrete event control approach to robotic assembly proved to be highly successful in detecting and recovering from undesirable contact states resulting from misalignment or mismatch between the model and the actual system. The system was able to recover from several situations of mismatch at several speeds. The ability of this system to recover from an otherwise damaging situation demonstrates a high level of performance and robustness for robotic assembly. It is one of the most important advantages of this method and significantly advances the state of the art in robotic assembly.

6 Limitations of the Approach

As with any system, certain limitations exist. The system proved to be very successful for robotic assembly. However,

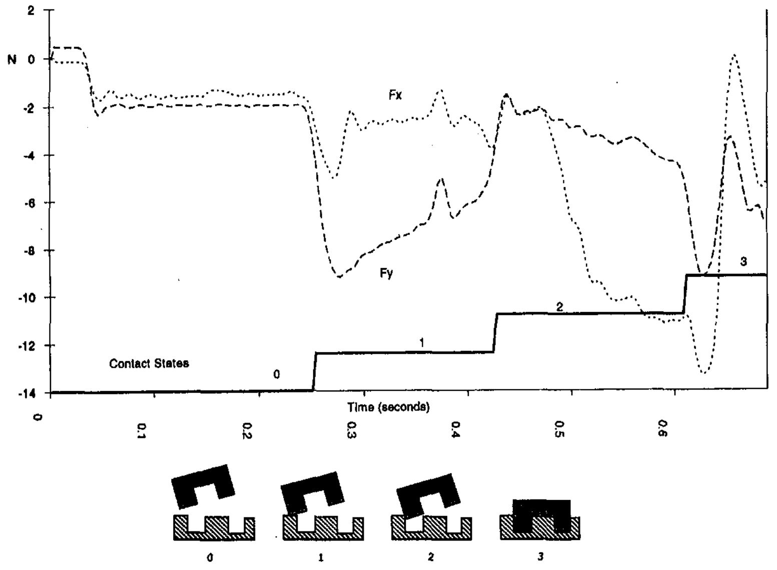

Fig. 13 Force profile and contact state recognition for contingency experiments—relative angle: 30 deg

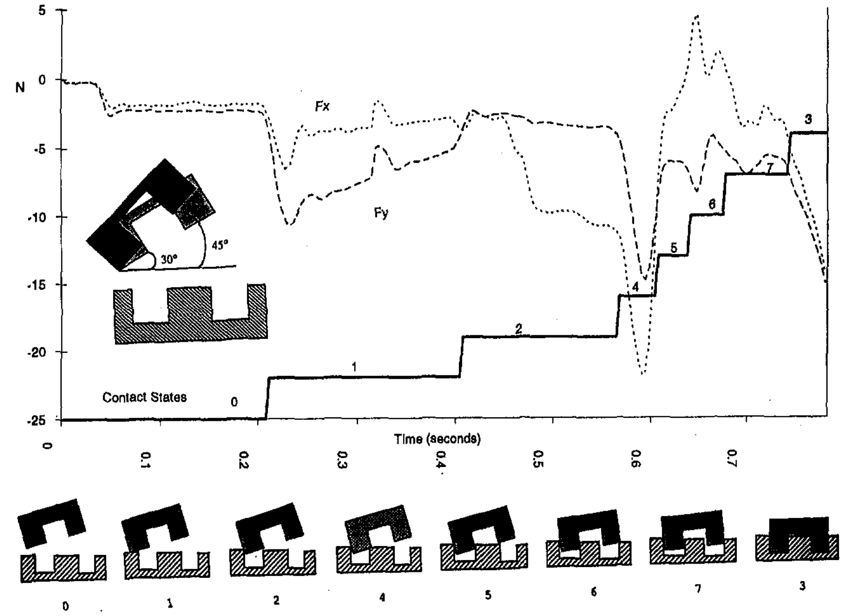

Fig. 14 Force profile and contact state recognition for contingency experiments—relative angle: 45 deg

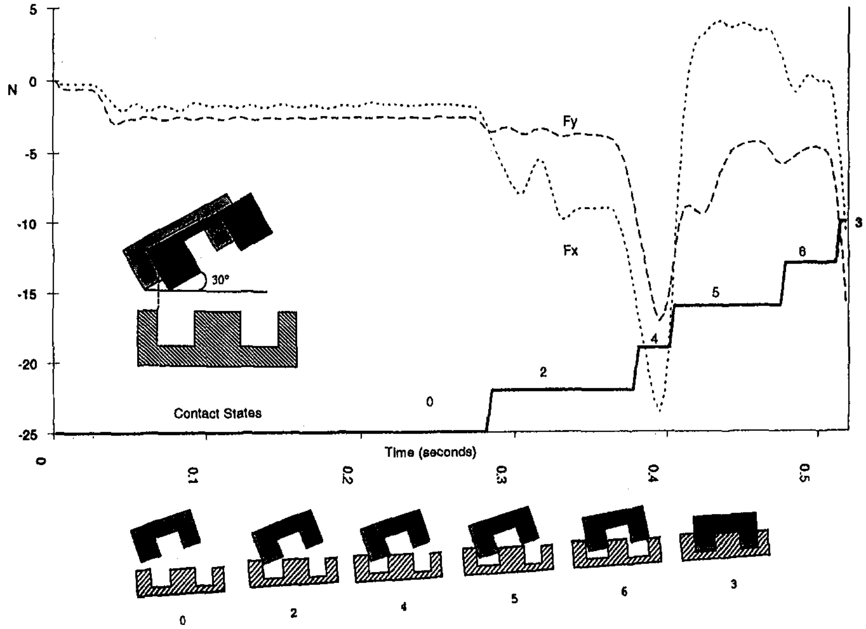

Fig. 15 Force profile and contact state recognition for contingency experiments—robot misalignment

there is a limitation within which application of the discrete event method is appropriate. The primary limitation to the discrete event control approach is the required processing. Since the constraints are functions of the state variables, the velocities need to be recalculated as the system moves. At very high speeds, the processing time required to recognize changes in state and to calculate and send the appropriate velocity commands is too long. The result is that the monitor will detect a change of state, and in the mean time, the system has transferred into another discrete state. The end product is a misaligned part. This situation was encountered at only the highest speeds of the robot 75 cm/s, where other factors, such as vibration, also played a part.

There are two solutions to this issue. The first is to run the system within the processing capabilities of the computer. If the highest speeds attainable are not acceptable, a more powerful computer should be used. The second approach is to minimize the amount of processing required. The velocity commands for most of the discrete states of contact can be determined off line and stored in memory. The process recognition can be streamlined with only the transition recognition implemented.

Note that problems with the discrete event controller were very rare. Problems only arose at the very highest of speeds. Often the problems were associated with the vibrations of the system or with the moving of the constraint geometry under the high impact forces. Even so, ill results did not occur frequently. Also, the system was easily restored by lowering the speed slightly to, say, 70 cm/s.

7 Conclusions

This paper presented a highly effective and robust approach to the control of robotic assembly. This new method modelled assembly as a discrete event system, where a discrete event was defined as a change in the discrete state of contact. Petri nets were used to highlight the key components of the control synthesis problem. Based on the specified discrete event trajectory, discrete control variables were derived. Each discrete control variable either enables or disables a discrete event. Using the discrete controls and the constraints that they represent, a set of simultaneous inequalities was derived. The solution of this set of inequalities is a velocity vector that will effect the desired discrete event. The solution of the inequalities was accomplished using the simplex method of nonlinear programming. The combination of the optimal trajectory, the optimal velocity commands and the Petri net model of assembly proved to be effective in recognizing and recovering from unplanned situations, showing a robustness to the method. This paper, in conjunction with [10], significantly advance the state of the art in robotic assembly.

In an effort to address some of the performance and robustness issues associated with a model-based approach to robotic assembly, extensive experimentation was conducted. The goal of the experimentation was first to prove the feasibility of both the dynamic model-based process monitoring and the discrete event control of robotic assembly. This was accomplished by monitoring a complicated trajectory and by commanding an optimal event trajectory. The discrete event control proved highly successful with a three-event trajectory executed in a very short time, 0.2 seconds.

The second goal of the experimentation was to show some of the performance and robustness characteristics of the overall robotic assembly system. The second goal was met by demonstrating the ability of the Petri net approach to successfully handle unexpected and undesired situations through contingency calculations. It was demonstrated that the system will recover from a mismatch between the model and the real components. It was also demonstrated that the system will recover when a sub-optimal trajectory is encountered due to positioning errors or errors in the velocity commands.

The last goal of the experimentation was to determine some of the limitations of the discrete event approach to assembly. It was discovered that a performance limitation exists. The discrete event control had difficulty at very high speeds because of the processing time required. A high speed limitation of 75 cm/s is required due to the processing time required by the discrete event controller.

References

1 Asada, H., and Hirai, S., 1989, ''Towards a Symbolic-Level Feedback: Recognition of Assembly Process States,'' *Proc. 5th Int. Symp. Robotics Res.*, Tokyo.

2 Michael J. Best, and Ritter, Klaus, 1985, *Linear Programming: Active Set Analysis and Computer Programs*, Prentice-Hall, Inc.

3 Draper, J. V., Wrisberg, C. A., and Blair, L. M., 1988, ''Measuring Operator Skills and Teleoperator Performance,'' *Proc. Intl. Symposium Teleoperation and Control*, pp. 341–349, Bristol, England.

4 Holloway, L. E., and Krogh, B. H., 1990, ''Synthesis of Feedback Control Logic for a Class of Controlled Petri Nets,'' *IEEE Trans. on Automatic Control*, Vol. 35, May.

5 Hannaford, B., ''Task Level Testing of the JPL-OMV Smart End Effector,'' *Proceedings of the JPL-NASA Workshop on Space Telerobotics*, pp. 371–380, JPL Publication 87-13, Pasadena, CA.

6 Hirai, S., Asada, H., and Tokumaru, H., 1988, ''Kinematic Analysis of Contact State Transitions in Assembly Operations and Automatic Generation of Transition Network,'' *Trans. Society of Instrument and Control Engineers*, Vol. 24, No. 4.

7 Mason, M., 1981, ''Compliance and Force Control for Computer-Controlled Manipulators,'' *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-11, No. 6.

8 McCarragher, B., and Asada, H., 1992, ''A Discrete Event Controller Using Petri Nets Applied to Robotic Assembly,'' *IEEE/RSJ International Conference on Intelligent Robots and Systems*, June.

9 McCarragher, B., and Asada, H., 1993, ''Qualitative Template Matching Using Dynamic Process Models for State Transition Recognition of Robotic Assembly,'' ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 115(2A), June.

10 McCarragher, B., and Asada, H., 1995, ''The Discrete Event Modelling and Trajectory Planning of Robotic Assembly Tasks,'' ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 117, Sept. 1995.

11 Ostroff, J. S., and Wonham, W. M., 1990, ''A Framework for Real-Time Discrete Event Control,'' *IEEE Transactions of Automatic Control*, Vol. 35, No. 4, Apr.

12 Peterson, James L., 1981, *Petri Net Theory and the Modeling of Systems*, Prentice Hall.

13 Raibert, M. H., and Craig, J. J., 1981, ''Hybrid Position/Force Control of Manipulators,'' ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 102.

14 Ramadge, Peter J. G., 1989, ''Some Tractable Supervisory Control Problems for Discrete-Event Systems Modeled by Büchi Automata,'' *IEEE Transactions on Automatic Control*, Vol. 34, No. 1, Jan.

15 Ramadge, P. J., and Wonham, W. M., 1987, ''Supervisory Control of a Class of Discrete-Event Processes,'' *SIAM J. Control Optimiz.*, Vol. 25, Jan.

16 Tadmor, Gilead, and Maimon, Oded, 1989, ''Control of Large Discrete Event Systems: Constructive Algorithms,'' *IEEE Transactions on Automatic Control*, Vol. 34, No. 11, Nov.

17 Whitney, Daniel E., 1987, ''Historical Perspective and State of the Art in Robot Force Control,'' *The International Journal of Robotics Research*, Vol. 6, No. 1, Spring.

18 Whitney, Daniel E., 1982, ''Quasi-Static Assembly of Compliantly Supported Rigid Parts,'' ASME Journal of Dynamic Systems, Measurement, and Control.